# Prediction of Airfoil Efficiency by Artificial Neural Network

**Chinmayi Siddamsetty, D. Jyothika, Tabindah Saleem, P. Manvitha, H. V. Srikanth, and S. Vijay Kumar**

## 1 Introduction

It is essential and critical to validate the aerodynamic properties of an airfoil in designing the fluid flow application and developing the optimal characteristics in this regard. The aerodynamic properties of fluid flow around an airfoil are accurately predicted by using the Reynolds-averaged Navier–Stokes (RANS) equations. However, this method has very elevated computational costs and very long progression intervals. Several researchers from the aviation industry have validated machine learning techniques for solving fluid flow problems that are less time-consuming and cost-effective. Although recent advances in computational power and efficiency have greatly reduced these costs, performing numerical simulations remains a time-consuming and computationally intensive task for many practical applications. As a result, it was necessary to shorten the computational time and cost for solving problems related to fluid flow.

Several studies, namely incompressible, steady-state flows, predict flow fields, and aerodynamic force coefficients of airfoils, were used to determine the characteristics and performance of the airfoil.

These diversified classes of neural networks have been consistently used to ascertain the visual imagery that these techniques incorporate deep learning techniques, and training can be done by extracting the unique features of an employed convolution layer. For explicit problems, an ANN model is adopted. When this model is used

C. Siddamsetty (✉) · D. Jyothika · T. Saleem · P. Manvitha · H. V. Srikanth
Department of Aeronautical Engineering, Nitte Meenakshi Institute of Technology, Bangalore, India
e-mail: schinmayi72@gmail.com

S. Vijay Kumar
Department of Mechanical Engineering, Nitte Meenakshi Institute of Technology, Bangalore, India

 
N. R. Shetty et al. (eds.), *Advances in Communication and Applications*, Lecture Notes in Electrical Engineering 1105, https://doi.org/10.1007/978-981-99-7633-1_1

for the same purpose, it assists in identifying solutions that can be easily validated (Narendra and Parthasarathy 1990; Hunt et al. 1992).

Ahmed and Kamal (2022) proposed that the ability of a BPNN model to predict the significant aerodynamic features Cl and Cd of airfoils was evaluated. The training was carried out for the BPNN model, validated, and tested using aerodynamic data obtained through numerical simulations of 440 cases. In summary, the BPNN produced promising results for predicting the aerodynamic coefficients of airfoils that were contemplated for distinct conditions. Moin and Khan (2021) cited the best geometric coordinates which were determined by training and comparing the network architecture performance of airfoil. In contrast to CFD analysis, the neural network was found to be proficient in capturing aerodynamic properties with sparse geometric information.

Sekar and Khoo (2019) carried out investigations without directly solving the Navier–Stokes equations, where CNNs were used to estimate the flow field over an airfoil as a role of airfoil geometry, Re, and more. Chen et al. (2020) created a dataset of CAI to determine the CL and CD using flow condition convolution. Bhatnagar and Afshar (2019) used CNNs to validate velocity and pressure fields. There are reports that CNN is used to explain the pressure distribution pattern around the airfoils and to predict the nonuniform study laminar flow in the 2D or 3D domain (Hui et al. 2020; Guo et al. 2016). However, simple ANN architectures have been extensively adapted for addressing the problems associated with designing of an inverse airfoil. Rai and Madavan designed turbomachinery airfoils utilizing pressure distribution (Rai and Madavan 2001) and other design variables (Rai and MAdavan 2000) as input to ANNs. Huang et al. (1994) used ANNs to design and evaluate the Eppler method, which explains the velocity distribution of the airfoil.

The authors claimed that the lift coefficients determined by this method are highly accurate in contrast to the claim, and it is observed that there was a deviation between predicted and actual coefficients. Khurana et al. (2008) used an approach with an ANN on PARSEC (a common method for airfoil parameterization) airfoils as a search agent in order to optimize their shapes. Various reports reported the possibility of using supervised machine learning techniques in the field of aerodynamics (Duru et al. 2021). However, in the previous studies, the ANN model was trained for lesser cases with less range of AoA and needs further investigations to prove its adaptability.

In this study, the test cases were increased along with the range of AoA. The numerical simulation was performed in CFD. This resulted in the model being more accurate in predicting the Cl and Cd for different airfoils. The ANN model was trained in MATLAB using a dataset with different operating conditions such as Re and AoA as input data and the corresponding CL and CD as output data. The primary goal of this research was to develop an ANN architecture capable of predicting the most important aerodynamic characteristics Cd and Cl, which were determined for different airfoils under various aerodynamic conditions.

## 2 Methodology

### 2.1 Numerical Simulation

The numerical simulation was performed in the CFD software package Ansys Fluent 16.0. The numerical simulations were performed on different airfoil geometries under various flow conditions to obtain the coefficient of lift and drag which then could be used as a training data set to train and validate the ANN model. The CFD analysis on four different NACA series airfoils, i.e., NACA 0012, NACA 0015, NACA 2412, and NACA 4415 was carried out. NACA 0012 and NACA 0015 are symmetrical airfoils and NACA 2412 and NACA 4415 are asymmetric airfoils that have various applications in the field of aeronautics. The simulation was performed for 20 different AoAs on each airfoil varying from $-10°$ to $10°$. The Re varied from $0.5 \times 10^6$ to $5 \times 10^6$, i.e., the Re used in the simulation was $0.5 \times 10^6$, $1 \times 10^6$, $1.5 \times 10^6$, $2 \times 10^6$, $2.5 \times 10^6$, $3 \times 10^6$, $3.5 \times 10^6$, $4 \times 10^6$, $4.5 \times 10^6$, and $5 \times 10^6$. Each airfoil was simulated with each Re at the AoA varying from $-10°$ to $10°$ with a $1°$ increment between the cases. The meshing details of selected airfoils are shown in Figs. 1a–d.

Altogether, the CFD analysis of 850 such cases was performed. The K-epsilon turbulence model was used which is a general description of turbulence by means of PDE. The standard k-epsilon model was used for a much more practical approach which helps in minimizing the unknowns and presents a set of equations that could be applied to various turbulent applications (Li et al. 2020). The C-type domain was used with the boundary conditions given as velocity inlet and pressure outlet at the inlet and outlet, respectively. The velocity specification was selected as magnitude and direction where magnitude was changed according to the Re. The velocity components x and y are changed with respect to the AoA as cos$\alpha$ and sin$\alpha$.

When an airfoil is subjected to flow, its streamlined shape enables the flow to split between the airfoil's upper and lower surfaces. The pressure in the upper surface of the airfoil decreases as the flow stretches over the curved upper surface, whereas in the flat lower section, the flow speed and pressure are constant. At the leading edge of the airfoil, the pressure distribution is maximum, whereas at the trailing edge, flow separation occurs. The static pressure distribution is predicted using CFD, and the contour plots are shown in Figs. 1e–h. The conditions for numerical simulations for airfoil flow analysis are mentioned in Table 1.

### 2.2 Feedforward Backpropagation Neural Network

A feedforward backpropagation neural network (ANN) is a form of ANN that has three layers: the input layer, the hidden layer, and the output layer. The substantial number of hidden layers is determined by the nature of the challenge. The number of neurons or nodes in the input layer is determined by the quality of the input characteristics, and the number of nodes or neurons in the layer that produces output

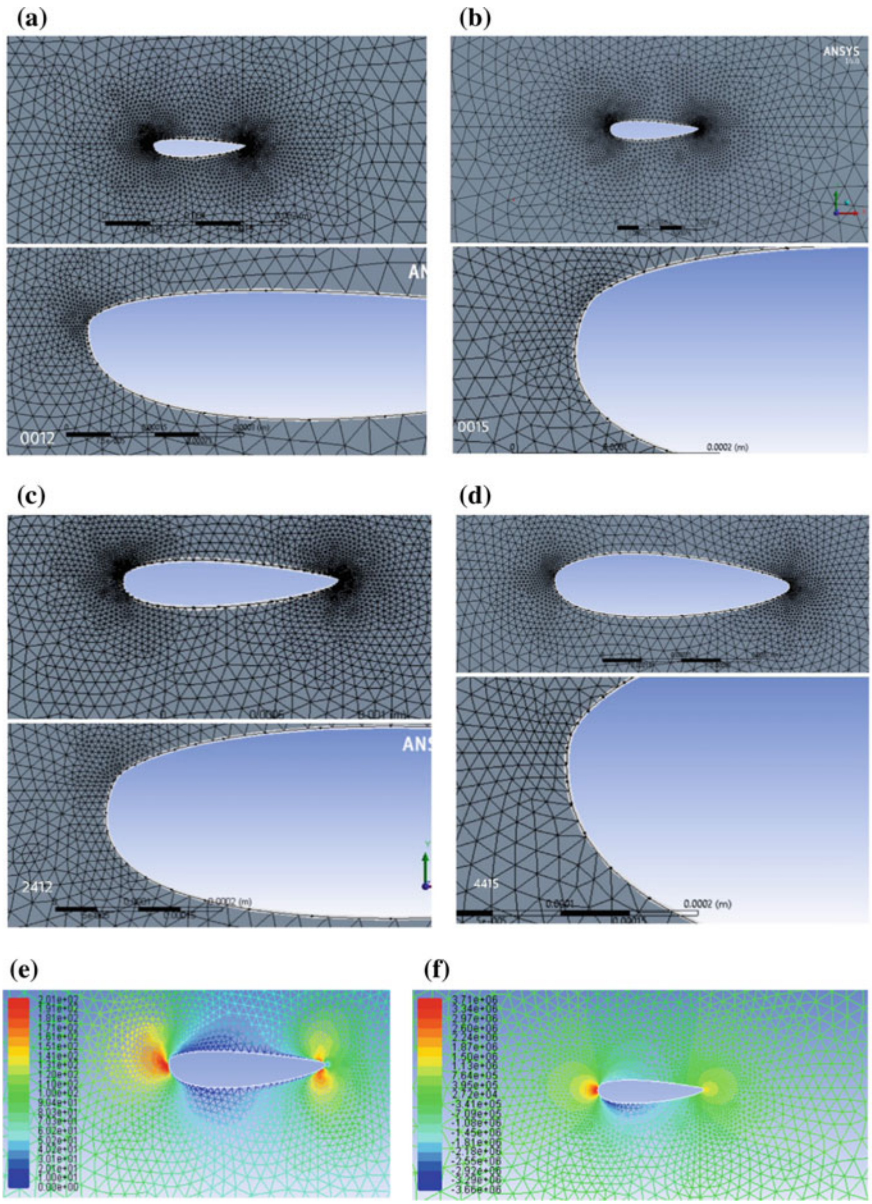


**Fig. 1** **a** Meshed airfoil NACA 0012, **b** meshed airfoil NACA 0015, **c** meshed airfoil NACA 2412, **d** meshed airfoil NACA 4415, **e** static pressure plot of NACA 0012, **f** static pressure plot of NACA 0015, **g** static pressure plot of NACA 2412, **h** static pressure plot of NACA 4415

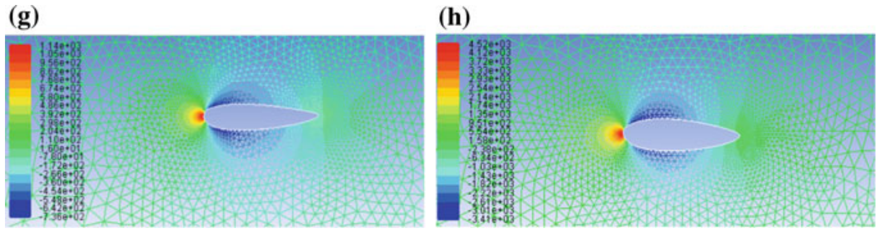


**Fig. 1** (continued)

is determined by the number of outputs. Weighted connections transmit all processing items to the next layer, such as neurons or nodes. The weights of the corresponding neurons have been multiplied through the hidden layer's inputs and aggregated by the hidden layer to generate a summed output. The output is then processed using the sigmoid, tan-sigmoid, or threshold activation functions (Yuan et al. 2018). This type of neural network is used because of its speed of functioning, accuracy, and due its ease of implementation. The general architecture of the model used is shown in Fig. 2.

The delta rule is known as the WIDROW-HOFF learning rule using a supervised learning technique. Our model was trained by the adaption learning function called gradient descent (LEARNGDM) which was adopted by the following procedure.

1. By giving all the random values each weight $w_{ij}$ and $b_j$ where $i = 1$ to $n$ and $j = 1$ to $m$.
2. Input and output datasets are fed into the BPNN model, and the output of each layer is calculated by the following equation:

$$y_{jp}^{[lc1]} = f\left(\sum_{i=1}^{N1} w_{ij}^{[l+1]} y_{ip}^{[l]} + b_j^{[l+1]}\right).$$

3. Error at the output layer is computed by the below formula:

$$\text{err}_{jp}^{[L]} = f\prime\left(y_{jp}^{[L]}\right)\left(d_p - y_{jp}^{[L]}\right),$$

in the $i_{\text{th}}$ hidden layer ($i = L - 1, L - 2, \ldots 1$).

4. The following equations are used to calculate the bias between the input and output layers and changes in weights

$$b_{ij}^{[l]}(n+1) = b_i^{[l]}(n) + \text{NG.err}_{jp}^{[l]},$$
$$w_{ij}^{[l]}(n+1) = w_i^{[l]}(n) + \text{NG.err}_{jp}^{[l]}.y_{jp}^{[i-1]}.$$

Error terms are back propagated into the neurons of the previous layer while calculating the changes in weights and biases.

**Table 1** Numerical simulation for various flow conditions

| Airfoil NACA | Reynolds number | Angle of attack | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0015 | 500,000 | – 10 | – 9 | – 8 | – 7 | – 6 | – 5 | – 4 | – 3 | – 2 | – 1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| | 1,000,000 | – 10 | – 9 | – 8 | – 7 | – 6 | – 5 | – 4 | – 3 | – 2 | – 1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| | 1,500,000 | – 10 | – 9 | – 8 | – 7 | – 6 | – 5 | – 4 | – 3 | – 2 | – 1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| | 2,000,000 | – 10 | – 9 | – 8 | – 7 | – 6 | – 5 | – 4 | – 3 | – 2 | – 1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| | 2,500,000 | – 10 | – 9 | – 8 | – 7 | – 6 | – 5 | – 4 | – 3 | – 2 | – 1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| | 3,000,000 | – 10 | – 9 | – 8 | – 7 | – 6 | – 5 | – 4 | – 3 | – 2 | – 1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| | 3,500,000 | – 10 | – 9 | – 8 | – 7 | – 6 | – 5 | – 4 | – 3 | – 2 | – 1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| | 4,000,000 | – 10 | – 9 | – 8 | – 7 | – 6 | – 5 | – 4 | – 3 | – 2 | – 1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| | 4,500,000 | – 10 | – 9 | – 8 | – 7 | – 6 | – 5 | – 4 | – 3 | – 2 | – 1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| | 5,000,000 | – 10 | – 9 | – 8 | – 7 | – 6 | – 5 | – 4 | – 3 | – 2 | – 1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 0010 | 500,000 | – 10 | – 9 | – 8 | – 7 | – 6 | – 5 | – 4 | – 3 | – 2 | – 1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| | 1,000,000 | – 10 | – 9 | – 8 | – 7 | – 6 | – 5 | – 4 | – 3 | – 2 | – 1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| | 1,500,000 | – 10 | – 9 | – 8 | – 7 | – 6 | – 5 | – 4 | – 3 | – 2 | – 1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| | 2,000,000 | – 10 | – 9 | – 8 | – 7 | – 6 | – 5 | – 4 | – 3 | – 2 | – 1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| | 2,500,000 | – 10 | – 9 | – 8 | – 7 | – 6 | – 5 | – 4 | – 3 | – 2 | – 1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| | 3,000,000 | – 10 | – 9 | – 8 | – 7 | – 6 | – 5 | – 4 | – 3 | – 2 | – 1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| | 3,500,000 | – 10 | – 9 | – 8 | – 7 | – 6 | – 5 | – 4 | – 3 | – 2 | – 1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| | 4,000,000 | – 10 | – 9 | – 8 | – 7 | – 6 | – 5 | – 4 | – 3 | – 2 | – 1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| | 4,500,000 | – 10 | – 9 | – 8 | – 7 | – 6 | – 5 | – 4 | – 3 | – 2 | – 1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| | 5,000,000 | – 10 | – 9 | – 8 | – 7 | – 6 | – 5 | – 4 | – 3 | – 2 | – 1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 2412 | 500,000 | – 10 | – 9 | – 8 | – 7 | – 6 | – 5 | – 4 | – 3 | – 2 | – 1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |

(continued)

**Table 1** (continued)

| Airfoil NACA | Reynolds number | Angle of attack | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1,000,000 | −10 | −9 | −8 | −7 | −6 | −5 | −4 | −3 | −2 | −1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| | 1,500,000 | −10 | −9 | −8 | −7 | −6 | −5 | −4 | −3 | −2 | −1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| | 2,000,000 | −10 | −9 | −8 | −7 | −6 | −5 | −4 | −3 | −2 | −1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| | 2,500,000 | −10 | −9 | −8 | −7 | −6 | −5 | −4 | −3 | −2 | −1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| | 3,000,000 | −10 | −9 | −8 | −7 | −6 | −5 | −4 | −3 | −2 | −1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| | 3,500,000 | −10 | −9 | −8 | −7 | −6 | −5 | −4 | −3 | −2 | −1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| | 4,000,000 | −10 | −9 | −8 | −7 | −6 | −5 | −4 | −3 | −2 | −1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| | 4,500,000 | −10 | −9 | −8 | −7 | −6 | −5 | −4 | −3 | −2 | −1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| | 5,000,000 | −10 | −9 | −8 | −7 | −6 | −5 | −4 | −3 | −2 | −1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 4415 | 500,000 | −10 | −9 | −8 | −7 | −6 | −5 | −4 | −3 | −2 | −1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| | 1,000,000 | −10 | −9 | −8 | −7 | −6 | −5 | −4 | −3 | −2 | −1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| | 1,500,000 | −10 | −9 | −8 | −7 | −6 | −5 | −4 | −3 | −2 | −1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| | 2,000,000 | −10 | −9 | −8 | −7 | −6 | −5 | −4 | −3 | −2 | −1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| | 2,500,000 | −10 | −9 | −8 | −7 | −6 | −5 | −4 | −3 | −2 | −1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| | 3,000,000 | −10 | −9 | −8 | −7 | −6 | −5 | −4 | −3 | −2 | −1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| | 3,500,000 | −10 | −9 | −8 | −7 | −6 | −5 | −4 | −3 | −2 | −1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| | 4,000,000 | −10 | −9 | −8 | −7 | −6 | −5 | −4 | −3 | −2 | −1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| | 4,500,000 | −10 | −9 | −8 | −7 | −6 | −5 | −4 | −3 | −2 | −1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| | 5,000,000 | −10 | −9 | −8 | −7 | −6 | −5 | −4 | −3 | −2 | −1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |

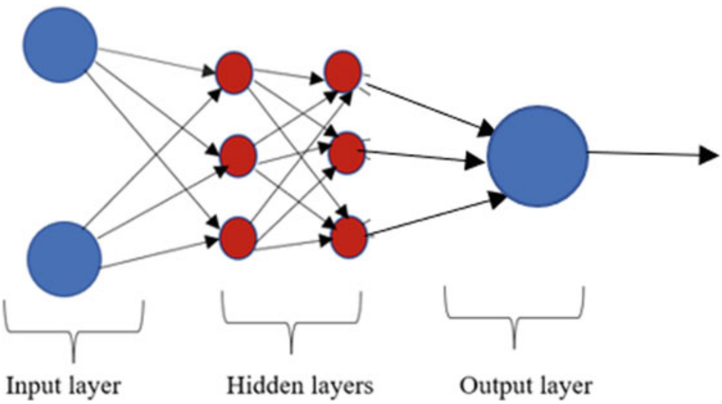


**Fig. 2** General block diagram of BPNN

5. Steps 2–5 are repeated till the errors are lesser than the minimum specified error.
6. The present study involves the usage of the sigmoid activation function and is given by the following equation:

$$F(x) = 1/\left(1 + e^{-x}\right).$$

## 2.3 *Prediction of Aerodynamic Coefficients*

The block diagram (Fig. 3) depicts the methodology used in this neural network which is used as a regression analysis tool to determine airfoil aerodynamic coefficients such as coefficient of lift and drag (Li et al. 2019). The dataset used in our model consists of 850 cases obtained through CFD analysis in the ANSYS fluent software on four different NACA airfoils: NACA 0015, NACA 0012, NACA 2412, and NACA 4415. Each of these NACA series was analyzed with ten different Re ranging from $5 * 10^5$ to $5 * 10^6$ for various AoA ranges.

The dataset contained two input parameters, velocity and AoA, as well as output parameters, Cl and Cd 70% of the data in the dataset was used for training the network, while 30% was used for testing and validation (Michos et al. 1983). Later, a neural network was trained using a backpropagation algorithm such as the Levenberg–Marquardt algorithm, and training continued until it reached a stopping criterion, which was based on the validation error, i.e., the error should reach a minimum value (Yildiz et al. 2415).

The architecture of the BPNN used consists of two inputs with ten neurons in the hidden layer and two outputs as shown in Fig. 4.

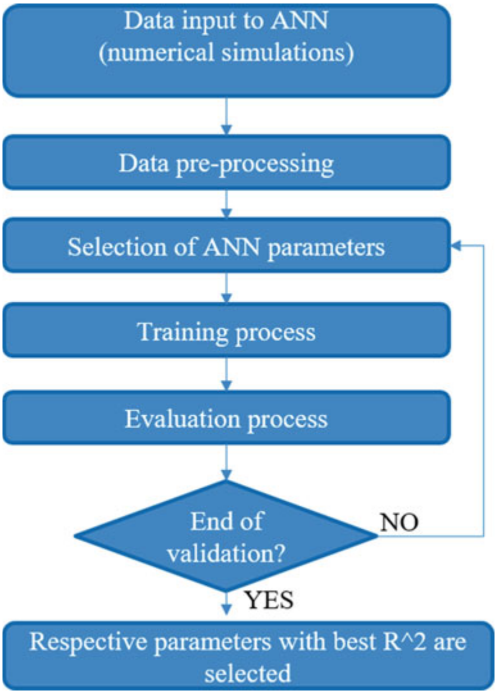


**Fig. 3** Flowchart of the steps followed

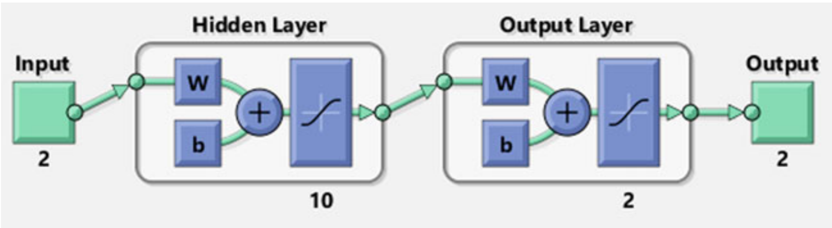


**Fig. 4** Architecture of BPNN model used

## 2.4 *Performance Evaluation*

The following parameters are used to evaluate the performance of the BPNN model.

1. RMSE: Lower the value of RMSE, higher the accuracy of the regression of the model, and it is calculated by the following formula:

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=0}^{n}|Pi - Mi|^2}.$$

2. Pearson correlation coefficient ($R$): $R$-value varies from − 1 to 1; the accuracy of the regression model is said to be high if it has $R$-value near to 1. $R$ I calculated by the following formula:

$$R = \sqrt{1 - \frac{\Sigma(Mi - Pi)^2}{\Sigma\left(Mi - \overline{Mi^-}\right)^2}}.$$

$n$ number of data points in the testing subset.
$P_i$ predicted values for the $i$th aerodynamic coefficient.
$M_i$ mean of all measured values for the aerodynamic coefficients.

## 3 Results and Discussions

The multiple numbers of neurons or nodes that are in the hidden layer are used to train the ANN model to get the best RMSE value. The RMSE values associated with number of neurons in the hidden layer are summarized in Table 2. Ten number of neurons in the hidden layer gave the best RMSE value, i.e., 0.0076318 at 27 epochs. It was discovered that the correlation ($R$) value was close to one for all training, validation, and testing cases indicating high accuracy of the predicted value. From the table, it may be inferred that changes in the number of neurons affect the RMSE value in zig-zag fashion indicating that RMSE value is sensitive to several neurons.

Figure 5 describes the relationship between the RMSE and neurons in the hidden layer. A minimum value of 0.0076318of RMSE with accurate prediction was achieved, but there was no continuous trend. As the number of neurons increases, RMSE value for neuron 10 has increased, and then, it decreased for neuron 14;

**Table 2** Correlation table for different neurons

| No. of neurons | Epochs | RMSE * $10^{-3}$ | Corr-coeff. for training | Corr-coeff. for validation | Corr-coeff. for testing | Corr-coeff. overall |
|---|---|---|---|---|---|---|
| *10* | 27 | 7.6318 | 0.97557 | 0.98447 | 0.98303 | 0.97812 |
| *14* | 24 | 6.5816 | 0.98402 | 0.98656 | 0.93697 | 0.97815 |
| *18* | 12 | 9.553 | 0.97581 | 0.97877 | 0.98238 | 0.97716 |
| *22* | 9 | 5.9927 | 0.97708 | 0.98695 | 0.97691 | 0.97839 |
| *26* | 6 | 9.5672 | 0.9747 | 0.98147 | 0.98265 | 0.97691 |
| *30* | 13 | 9.3398 | 0.98386 | 0.98061 | 0.94836 | 0.97769 |
| *34* | 13 | 10.746 | 0.98359 | 0.97696 | 0.95456 | 0.97792 |
| *36* | 5 | 11.265 | 0.97838 | 0.97377 | 0.99939 | 0.97757 |
| *40* | 7 | 8.5035 | 0.9727 | 0.98105 | 0.97739 | 0.97503 |
| *44* | 8 | 8.4694 | 0.9768 | 0.98165 | 0.9791 | 0.97809 |

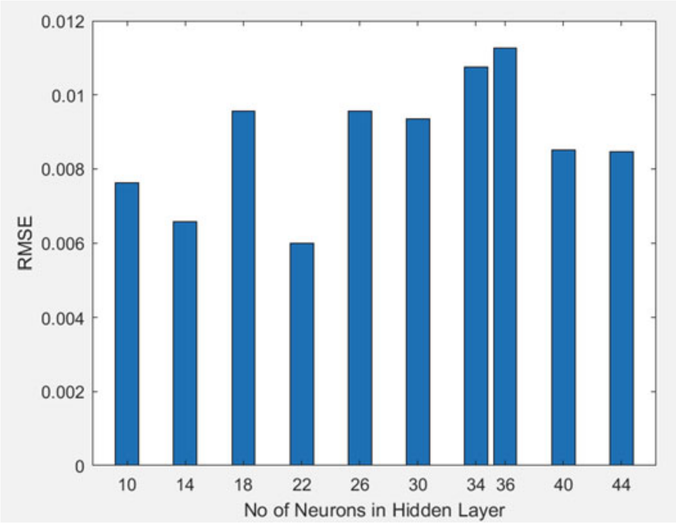


**Fig. 5** Neurons in hidden layer versus RMSE

later for neurons 18 and 22, RMSE value has increased and decreased respectively. RMSE value for the networks with neurons 26–36 was seen to increase, whereas RMSE value has again decreased with an increase in neurons to 44.

Figure 6 depicts the relationship between the RMSE value and the network used for the training, validation, testing, and prediction. The graph shows that it was proportionately high at the beginning of training and moderately decreased as training progressed. The RMSE value remained constant after 27 epochs at 0.0076318, and it did not decrease further. This performance graph only shows the best-case scenario. Compared to testing and validation data, RMSE value in the present study is large which may be due to the limited number of training data points in 850 instances.

From Fig. 7, we can conclude that the regression value for training, validation, and testing was near one which indicates high accuracy in the predicted values. BPNN model was trained and validated using the data which was obtained from the historical data. As each of the data points falls on the regressions line, the graphs demonstrate an excellent correlation between the target and projected values. In this work, the trained BPNN model indicates aerodynamic parameters with excellent accuracy and a relatively small number of inaccurate predictions when compared to other current models. Because machine learning techniques require a high number of data sets, model performance is directly related to data size. Only 850 simulated datasets were used to train the BPNN model in this investigation. Because 850 is such a small figure, it could indicate a constraint of the model's performance. To improve the model's dependability, more data from CFD simulations of various airfoils under various conditions must be collected. This will allow us to use this model to estimate aerodynamic coefficients of lift and drag for undiscovered airfoils under varied conditions.

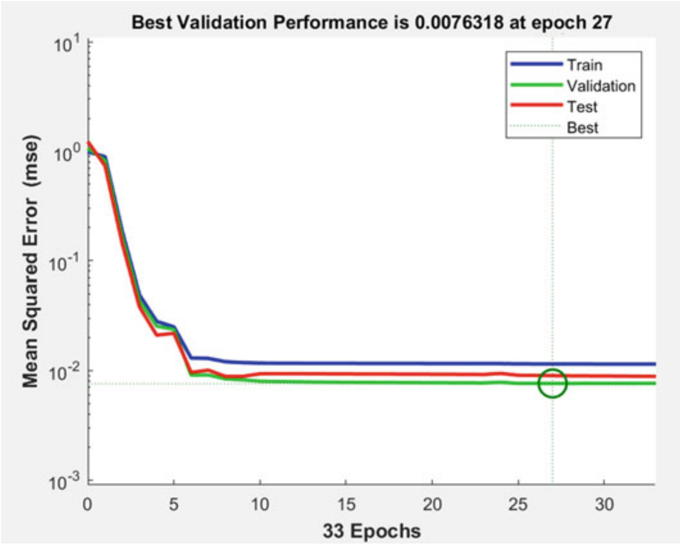


**Fig. 6** Performance plot for the best RMSE case

# 4 Conclusions

Validation of the BPNN model was done on basis of its efficiency to predict aerodynamic parameters. It is evident in Fig. 7 that there is a correlation between predicted values and the dataset as R is near to one. This research has contributed to the development of a model that can be used to determine the aerodynamic coefficients (CL and CD) of any airfoil and evaluate its performance. About 850 instances in the dataset obtained from numerical simulations are used to train, test, and validate the developed ANN model. Regression plots showed a nearly perfect fit between the actual and predicted values. At epoch 27, with ten neurons in the hidden layer, the RMSE for the best validation performance obtained was 0.0076318. The model supports any Mach number between 0 and 0.7 and any Re between $0.5 \times 10^6$ and $5 \times 10^6$. To summarize, the developed ANN model produced promising results in efficiently predicting the aerodynamic coefficients of various airfoils.

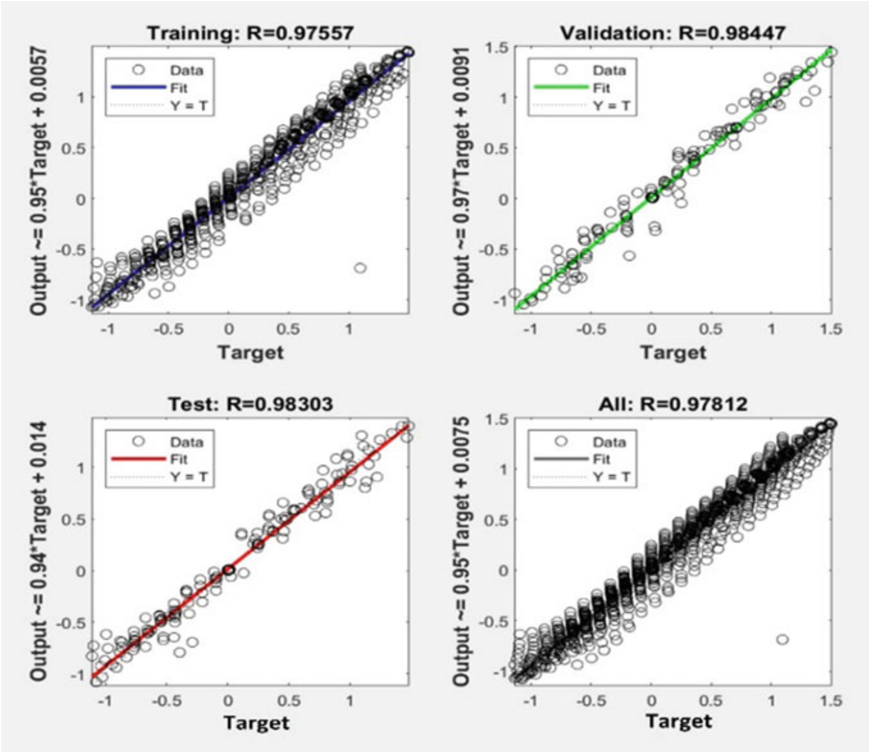


**Fig. 7** Regression plots for the best RMSE value

## Nomenclature

| | |
|---|---|
| ANN | Artificial neural network |
| Re | Reynolds number |
| M | Mach number |
| AoA | Angle of attack |
| Cl | Coefficient of lift |
| Cd | Coefficient of drag |
| Cm | Coefficient of moment |
| BPNN | Backpropagation neural network |
| RMSE | Root Mean Square Error |
| PDE | Partial Differential Equations |

## References

Ahmed S, Kamal K (2022) Aerodynamic analyses of airfoils using machine learning as an alternative to RANS simulation. National University of Sciences and Technology, Islamabad 44000, Pakistan. https://doi.org/10.3390/app12105194

Bhatnagar S, Afshar Y (2019) Prediction of aerodynamic flow fields using convolutional neural networks. Comput Mech 64:525–545. https://doi.org/10.1007/s00466-019-01740-0

Chen H, He L, Qian W, Wang S (2020) Multiple aerodynamic coefficient prediction of airfoils using a convolution neural network. Computational Aerodynamics Research Institute, China Aerodynamics Research and Development Center, Mianyang 621000, China. https://doi.org/10.3390/sym12040544

Duru C, Alemdar H, Baran U (2021) CNNFOIL: convolutional encoder decoder modeling for pressure fields around airfoils. Neural Comput Appl. https://doi.org/10.1007/s00521-020-05461-x

Guo X, Li W, Iorio F (2016) Convolutional neural networks for steady flow approximation. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining. https://doi.org/10.1145/2939672.2939738

Huang S, Miller L, Steck J (1994) An exploratory application of neural networks to airfoil design. In: 32nd aerospace sciences meeting and exhibit, p 501

Hui X, Bai J, Wang H, Zhang Y (2020) Fast pressure distribution prediction of airfoils using deep learning. School of Aeronautics, Northwestern Polytechnical University, 127 Youyixi Road, Xi'an, 710072, China.https://doi.org/10.1016/j.ast.2020.105949

Hunt K, Sbarbaro D, Zbikowski R, Gawthrop P (1992) Neural networks for control system

Khurana M, Winarto H, Sinha A (2008) Application of swarm approach and artificial neural networks for airfoil shape optimization. In: 12th AIAA/ISSMO multidisciplinary analysis and optimization conference

Li K, Kou J, Zhang W (2019) Deep neural network for unsteady aerodynamic and aeroelastic modeling across multiple Mach numbers. Nonlinear Dyn 96:2157–2177. https://doi.org/10.1007/s11071-019-04915-9

Li J, Zhang M, Martins JRRA, Shu C (2020) Efficient aerodynamic shape optimization with deep-learning-based geometric filtering. AIAA J 58(10):4243–4259. https://doi.org/10.2514/1.J059254

Michos A, Bergeles G, Athanassiadis N (1983) Aerodynamic characteristics of NACA 0012 airfoil in relation to wind generators. Wind Eng 7:247–262. https://www.jstor.org/stable/43749000

Moin H, Khan HZI (2021) Airfoils aerodynamic coefficients prediction using artificial neural network. Department of Aeronautics & Astronautics, Institute of Space Technology, Islamabad, 44000, Pakistan. https://doi.org/10.48550/arXiv.2109.12149

Narendra KS, Parthasarathy KK (1990) Identification and control of dynamical systems using neural networks. IEEE Trans Neural Netw

Rai MM, Madavan NK (2000) Aerodynamic design using neural networks. AIAA J 38(1):173–182

Rai MM, Madavan NK (2001) Application of artificial neural networks to the design of turbomachinery airfoils. J Propul Power 17(1):176–183

Sekar V, Khoo BC (2019) Fast flow field prediction over airfoils using deep learning approach. Phys Fluids 31:057103. https://doi.org/10.1063/1.5094943

Sun G, Sun Y, Wang S (2015) Artificial neural network based inverse design: airfoils and wings. Aerosp Sci Technol 42:415–428

Yildiz F, Turkmen AC, Celik C, Sarac HI (2015) Pitch angle analysis of NACA 2415 airfoil. In: Proceedings of the world congress on engineering, London, UK, 1–3 July 2015, p 5

Yuan Z, Wang Y, Qiu Y, Bai J, Chen G (2018) Aerodynamic coefficient prediction of airfoils with convolutional neural network. In: Zhang X (ed) Proceedings of the 2018 Asia-Pacific international symposium on aerospace technology (APISAT 2018), vol 459, Chengdu, China, 16–18 Oct 2018. Springer, Singapore, pp 34–46. https://doi.org/10.1007/978-981-13-3305-7_3